KAU Training Course

Intelligent Environments Training Package 1

Course Lecturer:
Vic Callaghan

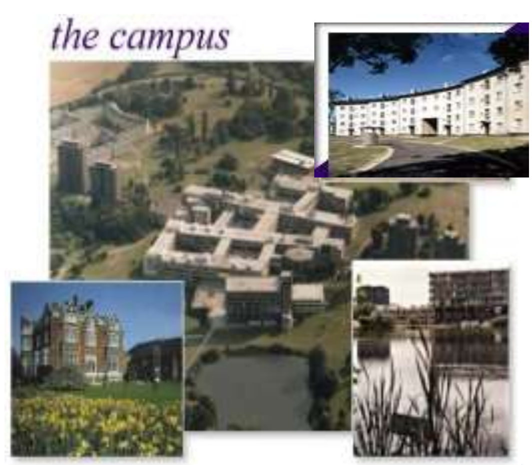
Lab Trainers:
Ahmed M H A Mohamed, Bo Yao

http://victor.callaghan.info

http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf

About Me

- Professor of Computer Science at Essex University
- Member of Intelligent Environments Group, Digital Lifestyles Centre & School of Computer Science and Electronic Engineering
- Expertise in intelligent environments artificial intelligence, Agents, Embedded-Computing, End-User Programming, Mixed Reality & Internet-of-Things.
- Part of organizational team for numerous conferences, workshops, journals



• Parkland of 200 acres
•Royal Charter in 1965
•11,939 students
•27% post graduates
•42% overseas (130 countries)
•Ranked 9th in UK for research

Some Activities I Organise

www.creative-science.org

CSF CREATIVE SCIENCE FOUNDATION

ICST* Transactions on Future Intelligent Educational Environments

*Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (ICST)

ICST.ORG

EAI European Alliance for Innovation

://icst.org/future-intelligent-educational-environments/

The Singularity Hypothesis (Volume 2): A Pragmatic Perspective Springer edited volume in The Frontiers Collection.

The Singularity – a point where AI transcends the limitations of peoples brains

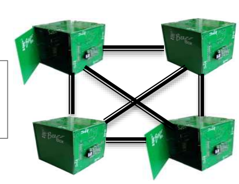
Structure – Training Packages

- Training Package 1 – **June 2014**
  - **Focus** – Understanding and building stand-alone intelligent environments

THIS IS THE TRAINING PACKAGE WE ARE FOLLOWING FOR THE NEXT 3 DAYS

- Training Package 2 – **February 2015**
  - **Focus** – Understanding and building connected (scaled-up) intelligent environments

- **Panelled construction facilitates arbitrary sized boxes.**
- **Boxes can contain multiple processors (agents)**
- **Environments can link to form multi-agent systems**



Training Package 1 – Aim

- The aim of this package is to provide an understanding of rule-based embedded-agent design techniques for constructing intelligent environments.
- In particular it aims to introduce:
  - the benefits,
  - architectural principles and
  - design methodologies

  associated with intelligent environments

Training Package 1– Learning Outcomes

After completing this package, you should be able to:

1) Explain the issues governing the specifications of embedded-agents and intelligent environments.
2) Explain the issues involved in the practical development of embedded agents and intelligent environments.
3) Design single-agent intelligent environments (from end-user managed to fully autonomous).
4) Apply intelligent embedded-agent techniques to the design of intelligent environments.

Training Package 1 – Outline Syllabus

- Day 1– Introduction (what are intelligent environments, technology, scenario, design issues)
- Day 2 – Embedded-Agents mechanisms for implementing reactive functionality
- Day 3 – Embedded-Agents mechanisms for implementing deliberative functionality
- Day 4 – Users, Programming-by-demonstration and Adjustable Autonomy

Training Package 1 – Lab Work

- Training Exercises
  - **Lab Exercise1** – Introduction to BuzzBox & RPi via means of a simple programming exercise to control the lights
  - **Lab Exercise 2** – Introduces the ideas of rule based reactive control agents based on programming a closed sensor-actuator loop to maintain a user set light level.
  - **Lab Exercise 3** – Invites the creation of a deliberative 'learning' agent' in the form of a Programming-by-Demonstration rule based embedded-agent that learns from the user.

Buzz Boards

Training Package 1 – Lab Work (1)

- Desktop Intelligent Environments (BuzzBoxes)



- Desk-top Intelligent Environments (eg mini smart-homes)
- Designed By ex-Essex PhD Student Victor Zamudio (in Mexico for his company FortiTo) Programmed by Marc Davies and Anasol Pena-Rios
- Upgradable with 30+ modules (including a mobile phone interface)

Training Package 1 – Lab Work (2)

- The development environment comprises a BuzzBox (the mini smart-home), display, keyboard and Raspberry Pi (RPi)

- Need ability to program in, Java
- Labs will provide introduction to embedded programming.

Reading & Information

- **Course notes** – are available to download from:
  - http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf
- **Lab Notes** (written by Marc Davies)
- **Lab Guidance** (provided by Ahmed M H A Mohamed, Bo Yao)
- **Research papers** (http://victor.callaghan.info/publications/)

Some Useful Papers

Papers– You will find a lot of useful papers at http://victor.callaghan.info/publications/

- **General Overview of Area** – Victor Callaghan, "*Intelligent Environments*", Chapter 5 in book called 'Intelligent Buildings' published by ICE Publishing, August 2013, available from http://victor.callaghan.info/publications/2013_IntelligentBuidings(IntelligentEnvironments).pdf
- **General introduction to embedded agents** – Callaghan V, et-al *"Programming iSpaces: A Tale of Two Paradigms*", in Springer-Verlag book Intelligent Spaces: January 2006 available from http://victor.callaghan.info/publications/2006_API06(ProgrammingISpacesATale).pdf
- **Introduction to Adjustable Autonomy** – Matthew Ball, Vic Callaghan, "*Managing Control, Convenience and Autonomy - A Study of Agent Autonomy in Intelligent Environments*", Journal of Ambient Intelligence and Smart Environments, IOS Press, pp 159 - 196, Volume 12, 2012, available from http://victor.callaghan.info/publications/2012_IOS12(ManagingControlConvenience).pdf
- **Introduction to Programming-by-Demonstration** – J. Chin, V. Callaghan, G. Clarke, "*End-user Customisation of Intelligent Environments*". In the handbook of Ambient Intelligence and Smart Environments, Springer, 2010, Spring, pp 371-407, available from http://victor.callaghan.info/publications/2009_JAISE09(EndUserCustomisation).pdf

Some Useful Videos

General – you will find a number of videos at http://victor.callaghan.info/videos/

- **iSpace** – an intelligent environment at Essex, (shown as part of EU project, Atraco) – http://www.youtube.com/watch?v=W2-wVlh4zFY
- **Embedded-Agents** – an old but informative look at embedded agent design http://www.youtube.com/watch?v=Ej3Hs4sC2DM
- **Adjustable Autonomy Agent** – introduction to variable intelligence http://www.youtube.com/watch?v=C_0yW6dQdZ4
- **Pervasive Interactive Programming** – demonstration of users programming smart spaces http://www.youtube.com/watch?v=TP56Y1Gh-3Y
- **Personal Operating Spaces** – an idea for mobile virtual spaces http://www.youtube.com/watch?v=AcFXeMDkTig
- **Blended Reality Desk** – Mixed reality using the desk in the new KAU lab http://youtu.be/U721DwHrwdY

Part 1

Overview

1. About Course (just did that)
2. What Are Intelligent Environments
3. Examples of Intelligent Environments
4. Living Labs
5. Embedded-Computing
6. Exemplar Scenario
7. Technical Issues (to be covered in course)

Living inside Machines

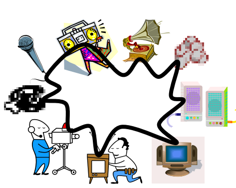
Le Corbusier, Villa Savoye, Poissy, 1928-31.

- Le Corbusier (1887-1965) famously remarked that, "*A house is a machine for living in".*

- "*A building is a robot we live inside"* *(Callaghan 2000)*

Machines Get Bigger – Intelligent Environments?

- They are: environments "where (**networked**) devices, services and applications work together seamlessly supporting even richer, more engaging and deeply connected (user) experiences" (Bill Gates)
  - Applications aim to design living & working environments that are more comfortable, usable, productive, secure, caring (medical), social, entertaining or energy efficient



- is **people** based and, to some extent, about **choice** (either unconscious or conscious) and **personalisation.**

- and is tied to nebulous concepts of social values and lifestyle

What Are Intelligent Environments?

- Could be a car, aeroplane, shop, office, home, planetary habitat etc
- An *Intelligent Environment* is an everyday living space in which numerous *network aware devices and services* combine to deliver information and *coordinate actions* in ways that enhance people's lifestyles

2 approaches to making intelligent environments

People Control

"A building is a machine we live inside" !

Agent Control

"An intelligent-building is one which contains automation of building management & control activities usually associated with needing human thought"

Terminology

- Alternative terms include:
  - ubiquitous computing,
  - pervasive computing,
  - ambient intelligence,
  - smart homes,
  - intelligent buildings
  - digital homes etc

Example of Intelligent Environment

BRE's Smart-Home (Integer House)

A collaboration between the UK's building Research Establishment and British Gas, to research and act as an exemplar for a range of issues that future homes might face like energy efficiency or using mobile phone app create and control of a set of network connected home systems

- The BRE Smart-Home (originally the Integer House) was created to develop an affordable, sustainable, intelligent and green future for housing in the UK. You can find out more by visiting http://www.bre.co.uk/news/Reinventing-the-house-of-the-future-909.html/

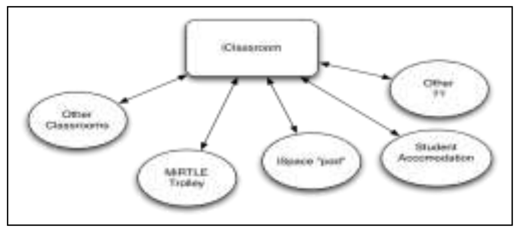
Example of Intelligent Environment

iSpace (at Essex University)

- Test-bed for ambient intelligent and pervasive computing in a domestic setting (Full sized 2 bedroom apartment)
- Sensor, actuator, computer and network rich environment to enable open-ended R&D
- Capable of supporting evaluations with long-term occupants

Example of Intelligent Environments
iClassroom

- An experimental high-tech classroom
- Designed to make maximum use of intelligent-agents to support all aspects of the teaching environment (environment, administration, learning)



Example of Intelligent Environments
Mixed-Reality Desk

The Essex-ID Immersive reality Desk

- New kind of student online learning experience based on mix of real video and avatars
- Transports real student into shared virtual environment (or vice-versa)
- Based on an idea from a method of using fiction to generate science ideas – "Science Fiction Prototyping" (see the link TalesFromAPod).pdf below)

http://www.immersivedisplay.co.uk

http://victor.callaghan.info/publications/2010_CS10(TalesFromAPod).pdf

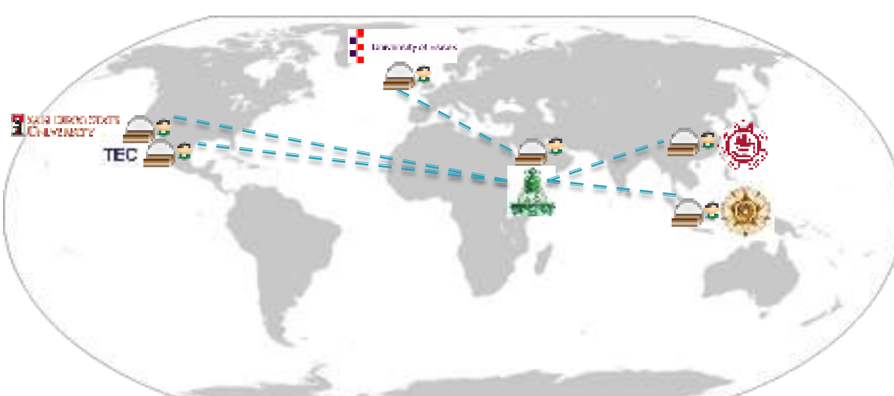
Example of Intelligent Environments
iCampus

HIPNet Project ***"Validation and Modelling of Next Generation Networks"***

Campus Coverage **(via WiMax Testbed)** **Suburb Coverage (5km radius)**

Essex University

EduNet



An international test-bed for teaching and research.

"the teaching Environment is the research"

Essex University

Intelligent Environments: Near-Term Vision

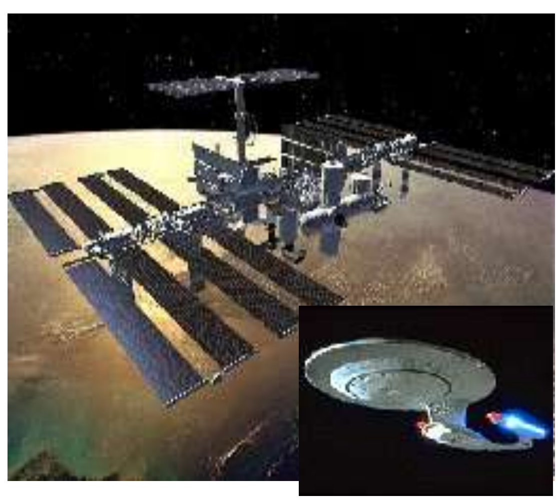
- Intelligent Interactive worlds built from *networked devices* that can be made to *coordinate* actions.
- Can *communities* of connected devices *self-program* themselves to deliver functionality people want?
- Can people design and create the functionality of their own "electronic spaces" (cf decorating spaces) or even make their own *virtual-appliances*.
- Solutions use *embedded-agents* and *programming-by-demonstration*, explained in this course


Intelligent Environments: Long-Term Vision

"Space, The Final Frontier!"

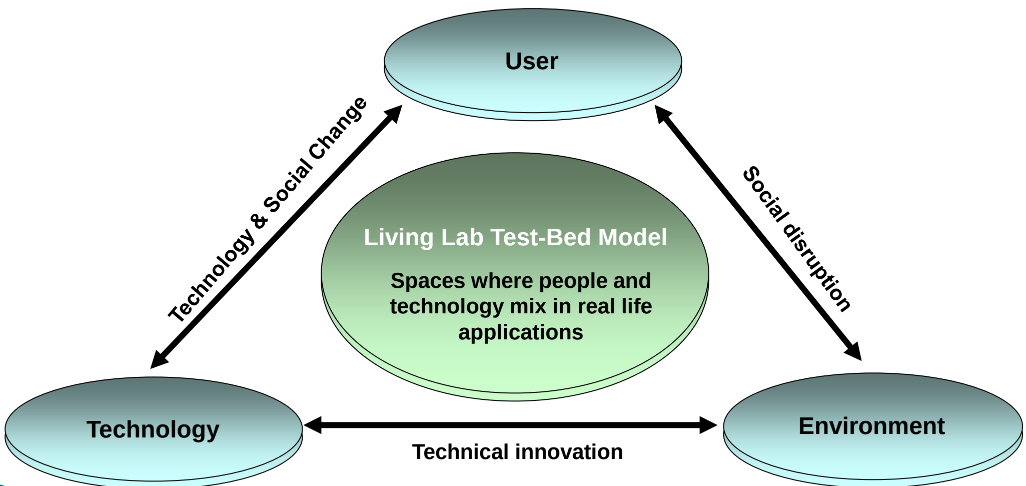
http://www.startrek.com/

Space Exploration
space vehicles,
planetary habitats
built from agent
based systems?



Living Labs –

Framework for Intelligent Environments Research



IE: An Illustrative Scenario – 1

"the 'radio-sphere' is awash with services available for use"

"monolithic appliances given way to deconstructed systems"

"interacts with the environment via a smart phone or pad"

- **Background** - *Liping is a visiting researcher at the University of Essex. She arrived at University and moved into her new temporarily accommodation, an intelligent apartment. Like all environments in the future the 'radio-sphere' is awash with services that are available for her use. Many are local services such as lighting, heating whilst others are remote services such as video, music, news, email. Monolithic appliances and computer applications have given way to more atomic networked functions (deconstruction) such as switches, video displays, codecs, editors, mp3 files etc. Liping interacts with the environment via her smart phone or pad which also contains descriptions of her preferred world.*

IE: An Illustrative Scenario – 2

- **Virtual Appliances & Applications** – *The concept of appliances and applications has lingered on as people still need to utilise functions akin to TVs, telephones, word processors etc. Consequently all environments had their networked devices / applications pre-formed into familiar default configurations (called Meta-Apps (MAps). Each MAp describes a familiar everyday appliance. Thus, both physical and information spaces functioned as normal. It is possible for users to purchase new MAps and, for more creative individuals, to devise their own.*

"deconstruct then reconstruct"

"Meta-Apps (MAps) describes a familiar everyday appliance or application"

School of Computer Science & Electronic Engineering
Essex University

IE: An Illustrative Scenario – 3

"if people move, their smart phone discovers what is available creating as near matches as possible"

"if devices move or fail, the system tries to find suitable replacements"

"sell missing devices"

- **Mobility** - *On entering her apartment, Liping's smart phone started to buzz in an unobtrusive manner indicating she was within a 'smart space'. Her smart phone contained her ontology based descriptions of her preferred MAps, discovered what was available in the environment and then requested as near matches as possible to be constructed. If devices moved or failed, the system would similarly try to find suitable replacements. Of course this was not always possible but her smart phone would indicate what was missing, so she had the option to borrow, buy or replace any missing devices. One such Map was her 'Communication Centre' (ComCen). On moving to other smart-spaces the smart-phone attempted to maintain Liping's preferred configuration for her ComCen MAp.*

School of Computer Science & Electronic Engineering
Essex University

IE: An Illustrative Scenario – 4

- **Programming** – *The original ComCen MAp consisted of a telephone service, audio transducer and dialler. Liping had modified the MAp to add in a light, video entertainment media stream and associated rules using a '<u>programming by demonstration' tool</u> that resided in her smart phone. For example, she had re-programmed the ComCen MAp configuration and rules to, "on receipt of a call, pause other incoming media streams, divert the call to the audio/video-transducer in use at the time, and raise the light if it is dark". While Liping generally only modified existing MAps there were numerous hobby clubs and small industries that generated novel and sometimes highly complex <u>MAps which they traded</u>.*

"Implicit autonomous agents or explicit user control"

"Trade in MAps"

IE: An Illustrative Scenario – 5

- **Interaction** - *Liping selects the 'News' menu, which causes the smart space to invoke an 'interactive display MAp', connecting it to her preferred RSS News feeds. Whilst reading her news feed, a video-conference request arrived, and the ComCen acted like a sophisticated '<u>soft-appliance</u>', activating previously programmed <u>rules</u> that caused the news feed to be suspended, lights to be raised and the video conference to be patched through to the current audio and video system. Like any appliance, Liping could manually override any of the settings on this "soft- appliance".*

"Users Rule, OK!"

"Interaction via wearable devices (phone?) or speech? etc"

Technology: Embedded Computers

Embedded-Computers – Putting computers into machines

An ***embedded-computer*** *is one that is integrated inseparately into a physical machine (eg a car, TV etc)*

Essex University

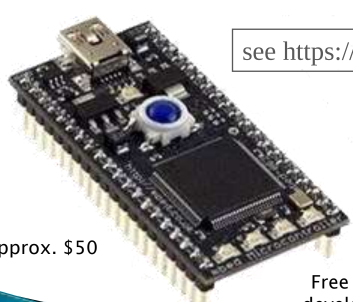
Technologies: Micro-Controllers

mBed: started by two ARM engineers 2005, as a way to developers (and students) easily prototype with microcontrollers

potential to use the internet to interconnect all computer based products (eg lighting, security systems)

- Original versions based around NXP (previously Philips Semiconductors) microcontroller, using ARM Cortex cores (S, M0 & M3) with latest version based on Freescale Semiconductor MKL25Z128VLK using ARM Cortex M4 core.
- Typical specifications are:
  - 96MHz,
  - 512KB FLASH,
  - 64KB RAM
  - various interfaces
    - Ethernet
    - USB Device,
    - CAN,
    - SPI,
    - I^2C

see https://mbed.org//



Approx. $50

Free open online development tools

Essex University

Technologies: Micro-Computers

- Arduino *
  - Based on AVR processor
  - Uses dated 5v devices (difficult to use with modern 3.3v hardware)
  - Expansion system based on modules called 'Shields' (some incompatibilities do to diverse developers).

- Raspberry Pi *
  - credit-card sized computer that plugs into TV & keyboard (uses Broadcom BCM2835 SoC based on an ARM1176JZFS core)
  - Fame derived from cost of $25 for cheapest version

*** Buzz-Boards work with all of these**

Buzz-Boards

Buzz Boards

Some Buzz Board modules

Examples: robot & Internet Radio

"Teaching Next Generation Computing Skills; The Challenge of Embedded Computing", Intelligent Campus 2011 (iC'11)

- Modular "embedded computing" teaching system (deconstruction /reconstruction)
- Desktop robot assembled using
  - ARM-Cortex mBed mezzanine,
  - Processor base board
  - Robot chassis (with IR proximity sensors and batteries)
- Internet radio assembled by plugging together
  - ARM-Cortex mBed mezzanine,
  - processor base board, network
  - keypad (optional)
  - audio *Buzz Boards*

www.FortiTo.com

BuzzBase

- Base-board accepts accepts other vendors modules & processors (eg mbed, RPi)
- Contains:
  - 8 General purpose push buttons with interrupt output
  - 8 tri-colour LED's
  - temperature sensor
  - light sensor (with a spectral response that matches the human eye)
  - audio sounder (that can also be used as a microphone),
  - high-resolution full colour OLED display
  - Both external DC and USB power operation
  - 2 bus ports that have I2C, SPI, and general purpose IO
  - 3-Axis accelerometer (optional)

Buzz Boards

www.FortiTo.com

BuzzBoard Hardware Modules

1. Mezzanine ARM
2. Mezzanine RPi
3. Processor Base *Buzz Board*
4. Audio-SD *Buzz Board*
5. Manual Control *Buzz Board*
6. Environmental Sensing *Buzz Board*
7. Navigation *Buzz Board.*
8. Inter-board Extension *Buzz Board*
9. Inter-board Right Angled *Buzz Board*
10. 3 Way Inter-board *Buzz Board*
11. Development *Buzz Board*
12. Prototyping *Buzz Board*
13. Keypad *Buzz Board*
14. LED Display *Buzz Board*
15. **Box** (panels) *Buzz Board*

1. Medical *Buzz Board*
2. MIDI *Buzz Board*
3. Network/232 *Buzz Board*
4. Quantum *Buzz Board*
4. RFID *Buzz Board*
5. Robot *Buzz Board*
6. Robot-Lite *Buzz Board*
7. Bluetooth *Buzz Board*
8. GPRS *Buzz Board*
9. WiFi *Buzz Board*
10. Range Finder *Buzz Board*
11. Aux Range Finder *Buzz Board*
12. Infrared Beacon *Buzz*
13. Battery *Buzz Board*
14. Test Point *Buzz Board*

Buzz Boards

www.FortiTo.com

Technology: Embedded-Agents

Embedded-Computers – Putting computers into machines

*An **embedded-computer** is one that is integrated inseparately into a physical machine (eg a car, TV etc)*

*In this course, we define intelligence as those problem solving activities we associate with needing human thought (eg **reasoning**, **planning**, **learning**)*

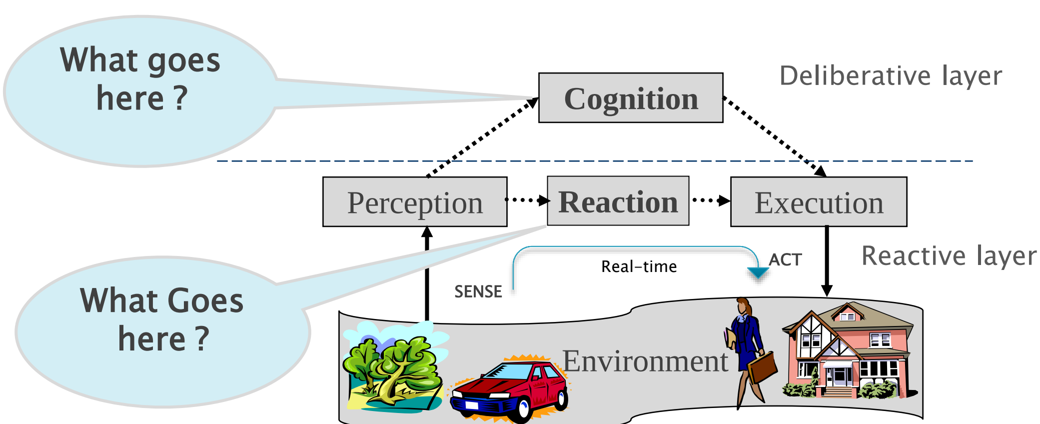
***Embedded-Agent** is an 'intelligent process' (eg **reasoning, planning, learning**) integrated into a computer based product*

an **embedded-agent** might be viewed as *"a knowledgeable and helpful friend embedded into a product which, by observation of your behaviour learns how to help"*

*"an **embedded-agent** can be regarded as a PID with reasoning, planning and learning"*

Embedded-Agents - Course Issues

The agent model we will use looks something like:



Design Issues: High Level Characteristics

- **Compact**: Small computational resources – a particular challenge for AI.
- **Reactive:** sense & respond in timely (ie **real-time**) fashion to changing environment.
- **Deliberative**: **Reason, plan & learn** in timely (ie **real-time**) fashion to changing environment in a **non-intrusive way** (learning doesn't interfere with occupant) and leaves "*the user as King/Queen*" (particularisation over generalisation)
- **Mobile**: operate in environments where people or objects move.
- **Social**: Coordinate actions with other agents (distributed AI) & people, to carry out tasks more effectively.

Design Issues: The Control Problem

- **Technology**
  - Small computational resources
  - Imperfect sensing/effecting
  - Real-time operation
- **Intractability**
  - Numerous input variables
- **Non-Determinism**
  - Dynamic, complex physical world
  - People & Agents in the Loop (this is the biggest challenge)

⇒ **Modelling** (as in PID, PLC, SMPA) difficult or impossible!

Covered in this course

Design Issues: The AI Problem

- Compact reasoning, planning & learning (AI usually needs large computers)
- Multi-agent cooperation / coordination (with large groups)
- **Dynamic** (and sometimes numerous), non-deterministic & ad-hoc agent/people assemblies
- Stimuli Focusing (assessing value)
- **Temporal** processing (inc past events)
- Particularisation versus generalisation
- **Non-intrusive** operation (with people firmly in control)

⟹ Conventional AI techniques too fragile (model dependent) and bulky!

Covered in this course

Essex University

43

Wider Issues

Covered in this course

- How do **people** interact with intelligent environments?
  - How are such systems managed or programmed (via autonomous embedded-agents or by end user programming)
- How does **end-user programming** work
  - Programming-by-demonstration, MAp, Ontologies
- How do **embedded-agents** function?
  - Reasoning, planning, learning mechanisms implemented?
- What **communication** infrastructure, messaging protocol & programming language is appropriate
  - IP, Lonworks, Java, Jini, KQML, FIPA, DIBAL etc
- How are embedded-agents **developed & debugged**
  - Single / multi-agent debugging

Essex University

44

Wider Issues

- What is the **agents granularity** (eg gadget, person vehicle room, building etc) ?
- Should there be embedded-agents that are able to **reason, plan, learn, co-operate** …… and if so how ?
- Does being **situated & embodied** have consequences to agent design (eg modelling the world & people) ?
- How can **multi-agents** coordinate actions (to get group synergy)?
- How are **spatial** and **temporal events** handled?

Covered in this course

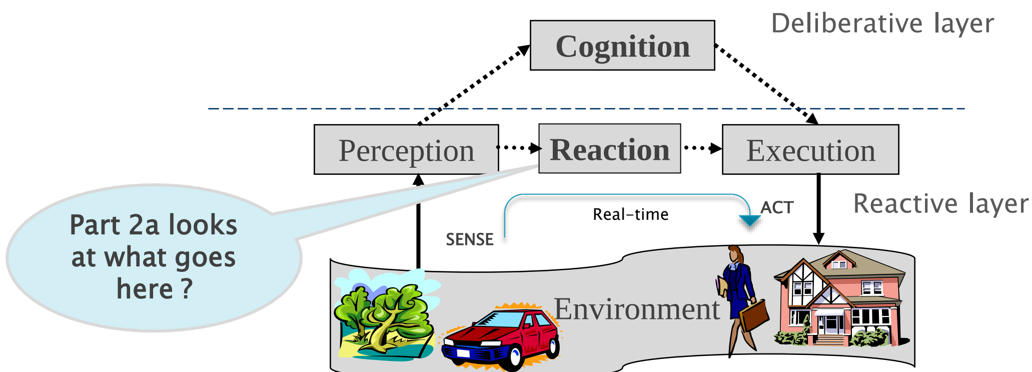
End of Part 1

That's it for Part 1 !

Part 2

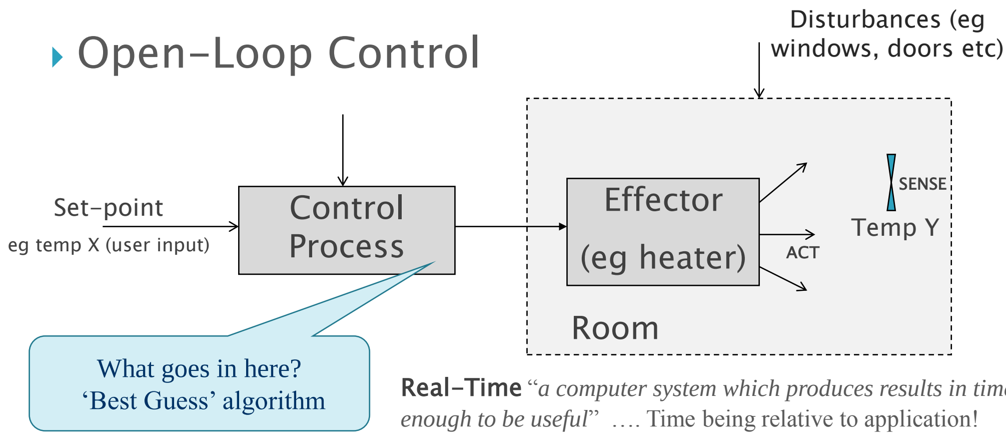
- Part 2a – The Reactive BBA Layer
  - Embedded Computing, Agents & Technologies
  - Brief review of control methods (eg PIDs, SMPA, BBA)
  - Behaviour based *embedded-agents*
- Part 2b – The Deliberative BBA Layer
  - The Role of memory
  - Deliberative Options
  - Evidential Learning

Embedded-Agents – Course Issues



Agent Options – Basic Automation 1

- Open-Loop Control

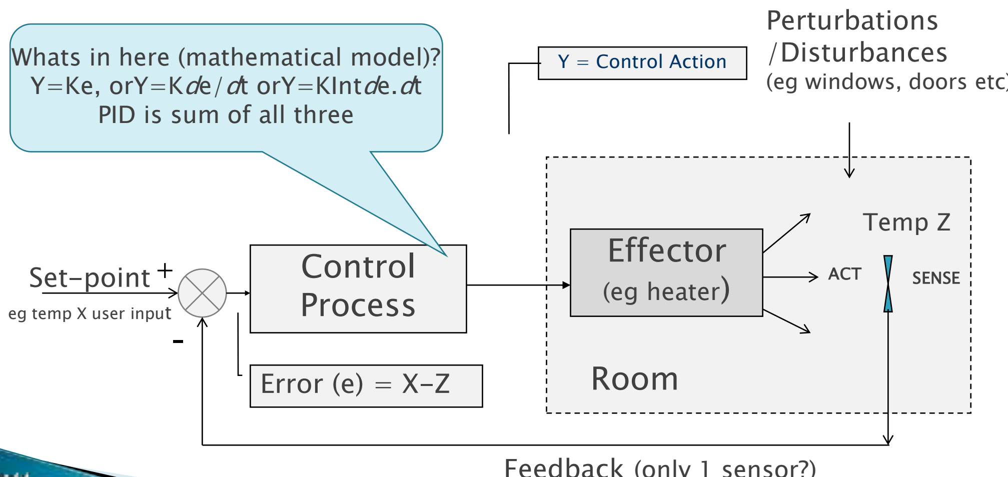


Real-Time *"a computer system which produces results in time enough to be useful"* …. Time being relative to application!

Real-Time *"a system whose correctness depends not only on the logical results of computation, but also on the time at which the results are produced"*

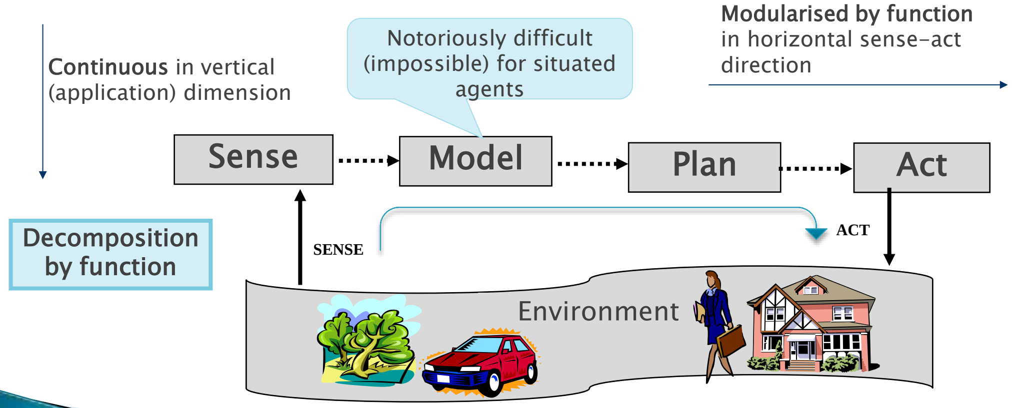
Agent Options – Basic Automation 2

- Closed-Loop Control



Agent Options: AI & SMPA Model

Simplified high-level view of
sense-model-plan-act scheme:



Deigning Embedded-Agents: The Model Problem

- Models are used extensively in Science & Engineering (and underpin most control systems)
- Traditional Control uses mathematical descriptions based on a linear <u>model</u> (eg differential equations)
- Traditional AI frequently uses an SMPA <u>model</u>

- Can we develop a suitable model for intelligent-buildings?
- Yes, for some individual subsystems (eg temperature controller) but ……….

Deigning Embedded-Agents: The Model Problem

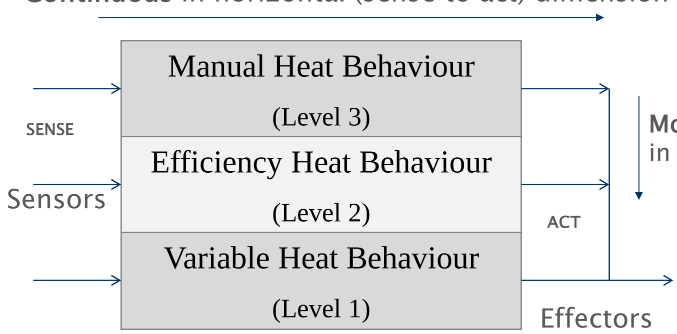
- How about the IE system as a whole ? Developing a model is difficult when it involves:
  - Potentially massive ad-hoc arrangements of highly dynamic real-time embodied and situated devices: ie tens of agents, hundreds of sensors (providing data of varying precision & accuracy) involving configurations liable to continual change and prone to failure.
  - complex natural phenomena (eg people & their idiosyncratic behaviour, the weather etc)
- Who can write a program to simulate or model a person (eg are people deterministic) ?

Deigning Embedded-Agents: The Model Problem

- Are there solutions to "*Model Problem*" ?
  - Yes "*the world is its own best model*" (Rodney Brooks, MIT)
- Brooks proposed a solution which used the world as its only model called *behaviour based architecture* (used for robotics work)

Agent Options: BBA (smart thermostat)

Continuous in horizontal (sense to act) dimension



decomposition by behaviour

Modularised by behaviour in vertical direction

- each level capable of generating a behaviour
- may be conflicts in behaviour requiring arbitration

Sometimes described as Horizontally *Layered* Architecture

Behaviour – "*a law for attainment or maintenance of goals*"

Embedded-Agents – BBA (smart thermostat)

Behaviour Based Architecture (BBA) works well in robotics which can be seen to have many similar characteristics to IEs and embedded-agents

Sometimes called subsumption architecture

In previous example:

- **Level 1 (adjustable)** – behaviour enabling 'heat engine' to "wander" around a 'temperature landscape' without hitting boundaries (ie safe min & max)
- **Level 2** (**energy-efficiency**) – behaviour to "steer" heat to efficiency goal (eg unoccupied = off) (subsumes level 1)
- **Level 3** (**manual-control**) – "Switch-following" behaviour (subsumes level 2). As operated by "willed" thoughtful person could be regarded as another "deliberative" layer.

Embedded-Agents - BBA

- Intelligence in reactive behaviour based system rooted in:
  - ensemble of simple competing behaviours interacting directly with world. Interplay produces pseudo reasoning & planning (ie intelligence)

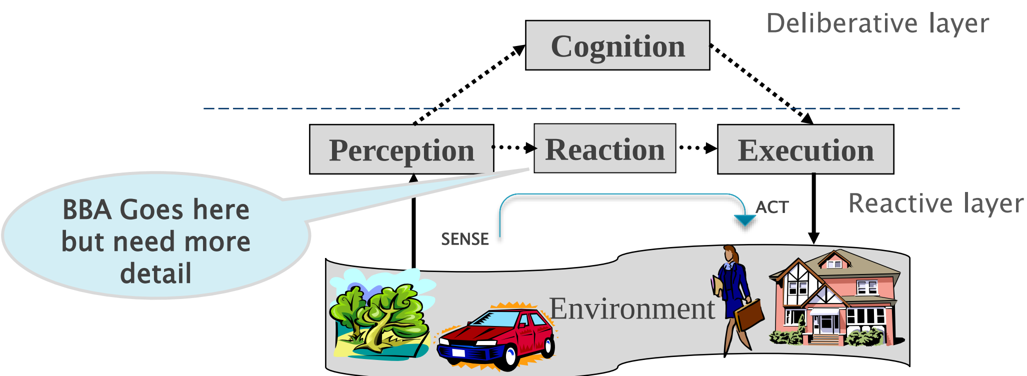
Note: Whilst overall behaviour is emergent, each sub-behaviour is deterministic and provides guaranteed safe actions (eg not bumping into obstacles)

  - Co-operating Behaviours Mechanisms (behaviour arbitration):
    - Switched Activation; combination of series of temporal actions (eg see Brooks subsumption)
    - Continuous Activation; combination of series of concurrent actions (eg see Steels, **Hagras**)

Environment context used to switch behaviours

Embedded-Agents - The Story So Far

Our agent model looks something like:



Robots ->Embedded-Agents ?

- Embedded Agents face similar problems to mobile robotic agents, they both:
  - use computationally small devices
  - are situated and embodied (interact with physical world)
  - operate in intractably complex environments where adequate models have proven impossible to provide
  - navigate the same "sensory space" → s-maps

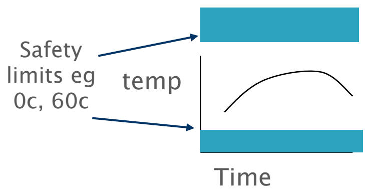
Agent Mobility: S-Maps

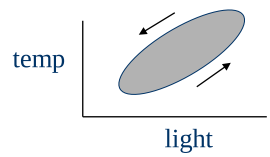
- Buildings (& all embedded-agents) can be viewed as "moving" through an abstracted perception domain referred to as a ***sensory map (s-map)***.
- "*s-maps*" contain "*distinctive features*" which correspond to landmarks or dynamic objects in Euclidean space (ie that navigated by robots).
  - Landmarks such as boundaries of world (eg temp limits)
  - Dynamic objects such as sudden temp drop (eg window being opened)

"Buildings are robots we live in" (Callaghan 02)

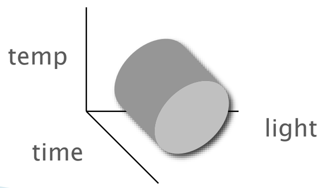
Agent Mobility: S-Maps

S-Map diagrammatic representation





For example, as the sun rises it gets light and the temp rises, as sun sets then the light falls and the temp follows, but lags.



Multi-dimensional s-map which I can't draw!

(apologies for poor drawing skills!)

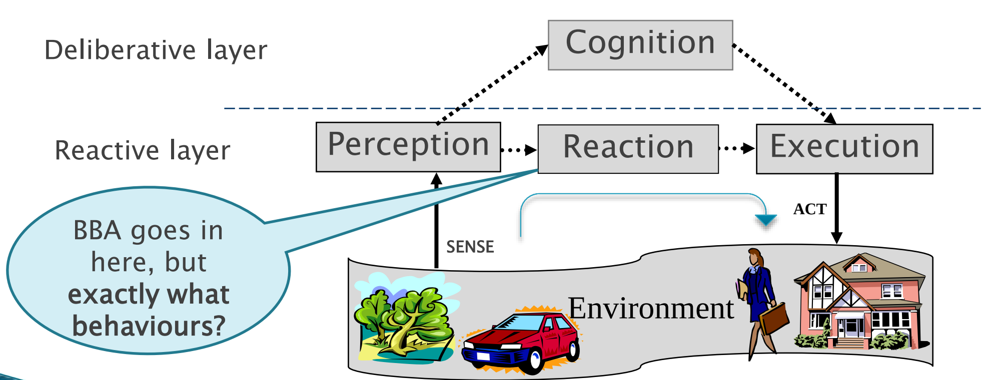
S-Maps & Behaviour Activation

- Encountering '*distinctive features*' (discontinuities) causes switch between different behaviours (providing pseudo reasoning & planning)
- Examples of '*distinctive features*' might be walls (for robots) or heat-barriers (for building agents) or sunrise/set for temporal dimensions (eg *distinctive-times*).
- A typical day can be represented as a signature (eg temp-light map on previous slide), deviations from which can be made to trigger alarm (abnormality detection – so called, "*inverted reasoning*")
- *S-Map* concept further reinforces application of mobile robot methods to IEs (and other embedded-agents)

Agent Options– The Solution

- Thus, as a consequence of the arguments on previous slides, intelligent-environment embedded-agents are commonly built from:
  - Behaviour based architecture (BBA)

Embedded-Agents: The Story So Far



IE Agents – A Simple Thought Exercise

- For example, why does a person adjust the temperature setting? Does it depend on:
  - heat, light, time of day, amount of physical activity, previous actions etc?
- If it does, these parameters need to be sensed and supplied to agent so it can learn why actions happened or reason as to what to do next !

Embedded-Agents: What Behaviours?

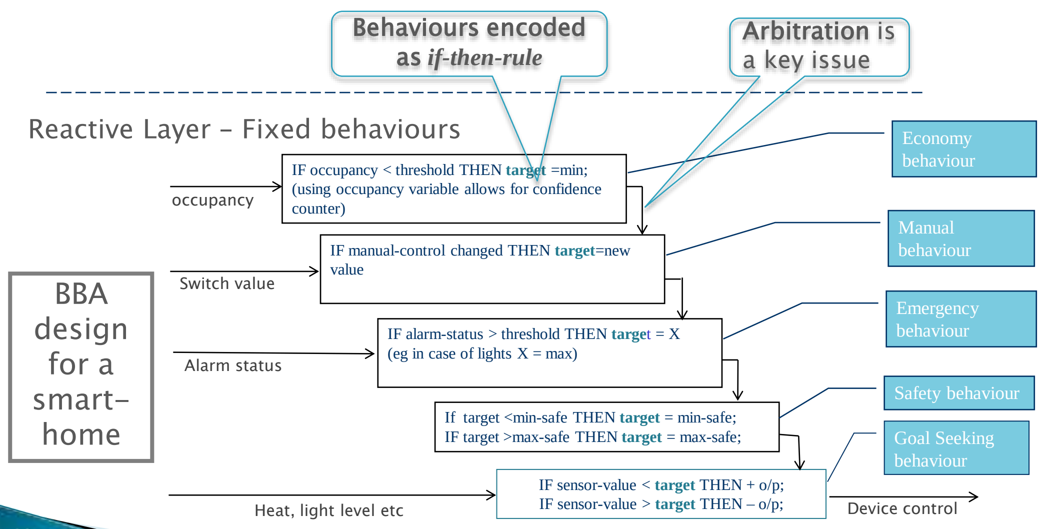
- To date no formal approach to deducing a optimum set of behaviours (eg a minimal set)
- There are numerous empirical guidelines
- A widely used approach is to note the choice of behaviours equates to guaranteed system actions (the overall system behaviour is emergent, but individual behaviours have inbuilt guarantees such as robots not hitting obstacles.
- **Guaranteed Behaviour Design Approach (guard)** – identify behaviours system needs to guarantee, and use them.

IE Embedded-Agents – What Behaviours?

Note: Guard guarantees system never strays outside safe bounds

- For most IE agents Fixed (Reactive) Behaviours formed from:
  - **Economy Behaviour** – responsible for conserving energy wherever possible
  - **Manual Behaviour**– ensures the agent is always responsive & subservient to the user; an IE agent axiom (and allowing the building to be at least as competent as one without the agent).
  - **Emergency Behaviour** – specifies what must happen in an emergency (sometimes combined with safety).
  - **Safety Behaviour** – prevents controlled quantity from going outside some limit.
  - **Goal Seeking Behaviour** – steers the virtual movement of the agent, through the s-map, to the target

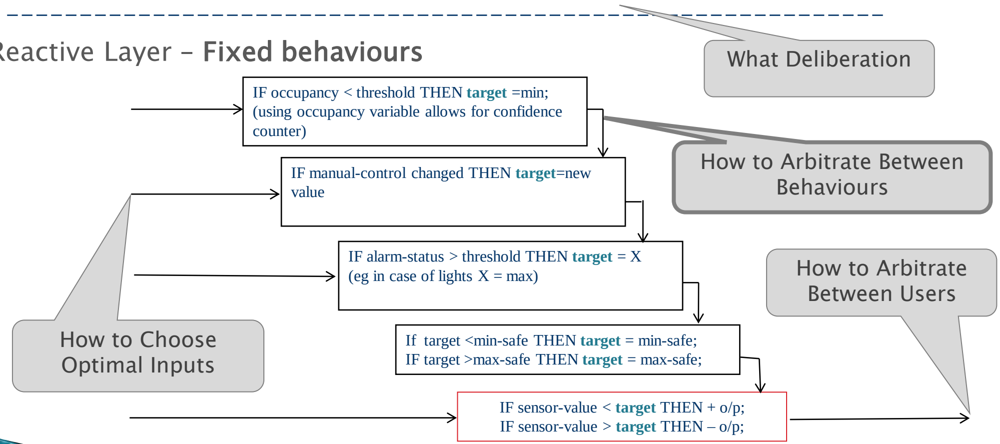
Example – A Possible BBA Design for Reactive Level



Other BBA Design Issues

Arbitration key issue (type: continuous/switched, by what: automatic/willed)

Reactive Layer – **Fixed behaviours**



Essex University

69

Embedded-Agents: What sensors ?

- Who Decides which sensor are appropriate for any given task?
- Traditionally it is the human designer but, for example
  - can he really predict what sensor set is most effective (eg may there be subtle changes in the environment that are important but he has missed?)
  - How can one deal with mobile agents (eg where sensor data may vary as different agent-worlds are encountered)
- One "school of thought" concludes "*agents know best*" (think about this and form your own view)

Essex University

70

Arbitration Mechanisms

- **Between Behaviours**

  Mechanism to determine which behaviour should have more influence on its action at a given instance, uses schemes such as:
  - Switched (Brooks)
  - Continuous (Hagras)
- **Between Users**

  Agent satisfying conflicting user needs at same instance, uses activation functions such as
  - spikes,
  - averaging etc

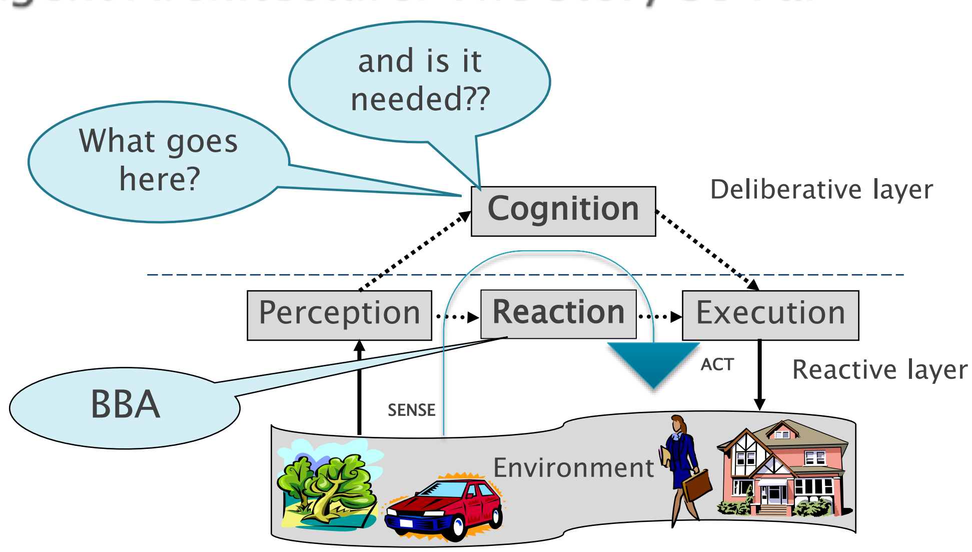
End of Part 2a

That's it for Part 2a !

Part 2

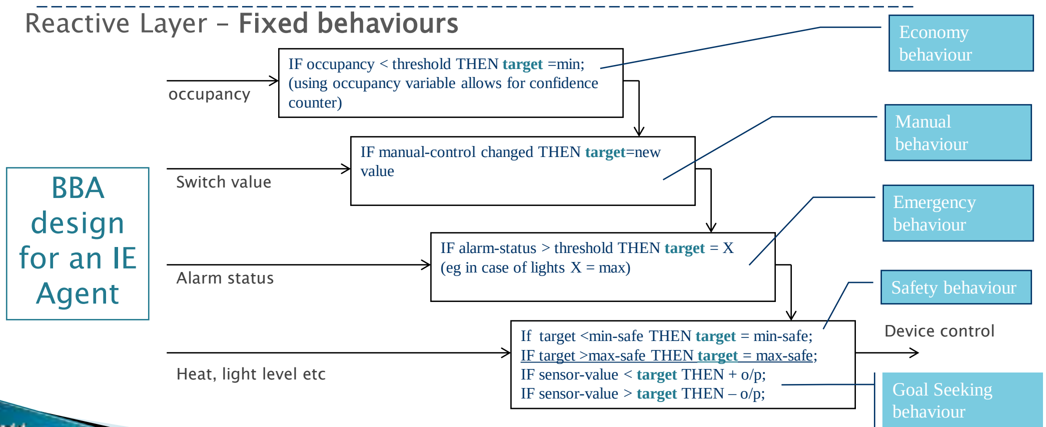
- Part 2a – The Reactive BBA Layer
  - Embedded Computing, Agents & Technologies
  - Brief review of control methods (eg PIDs, SMPA, BBA)
  - Behaviour based *embedded-agents*
- Part 2b – The Deliberative BBA Layer
  - The Role of memory
  - Deliberative Options
  - Evidential Learning

Agent Architecture: The Story So Far



Alternatively .....

What goes in here & is it needed – can behaviours be learnt (rather than fixed)?



Is A Deliberative Layer Necessary?

- **Question** – Can intelligent behaviour be obtained without cognitive feedback (ie could it emerge as an implicit outcome of unplanned temporal sequences of reactive behaviour) ?

cf. Sharkey's "environment strings" argument

- **Controversy** – *"There is some controversy surrounding the issue of architectures for intelligent machines. This controversy is sometimes referred to as the cognitive versus reactive architecture controversy"*
- **Answer ?** – One way of resolving this might be a form of Turing test based on behaviour … but what test?

The Role of Memory

- Reactive
  - Seems we can argue some reasoning & planning like behaviour does arise from interactions between reactive behaviours
  - But can there be there be an "intelligence" that is sufficient for IE embedded-agents without memory ?

Some interesting films have explored the role of memory in intelligent agents (us!): Total Recall, Memento, Eternal Sunshine of the Spotless Mind

- Deliberative
  - Learning implies memory (no memory in pure reactive systems)

Reactive or Deliberative Agents?

- *"although behaviour-based systems view knowledge representation as an impediment to effective and efficient control, it is still required for all but the simplest of intelligence".*
- We argue some deliberative capability in IE is necessary (so the agents can particularise and personalise their actions)
- Key issues
  - how to integrate ***reactive*** and ***deliberative*** components
  - how could deliberation work
  - Needs to be minimal (so need to avoid unnecessary overheads)

Embedded-Agents - Deliberation Options

- Majority of embedded-agents utilise a *multi-layered* architecture
  - reactive system) in the lower layer (eg pseudo reasoning, planning
  - deliberation system in the higher layer (eg learning, planning), occasionally with extra layers in between.
- Two main approaches to deliberation in BBA IE embedded-agents:
  - Dynamic reconfiguration of reactive control
  - Dynamic spawning of behaviours

Embedded-Agents - Deliberation Options

- Dynamic reconfiguration of reactive control
  - In effect, works by managing behaviour arbitration. Two main modes of operation:
    - ***automatic***, - consists of multiple parallel activity threads (schemas, behaviours),
    - ***willed***, - change focus of attention by altering threshold values for schemas dynamically
- Dynamic spawning of behaviours
  - In effect, work by recording actual user behaviour, then encoding this as a software process (eg rules) to reproduce original behaviour; a dynamic behaviour

Dynamic Reconfiguration of Reactive Control

- **Automatic** (most common in compact agents)
  These generally use distinctive feature based switching principles (eg Kuipers & Byun). Common forms are:
  - Switched Activation (eg Brooks subsumption)
  - Continuous Activation (eg Steels, Hagras)
- **Willed** (most common for joining BBA and Traditional AI)
  Work by managing activation of behaviours. Typical managers can be:
  - Traditional AI based planner (ie way of integrating BBA and traditional AI)
  - Person (eg via HCI)

Dynamically Spawning Behaviours

- Principle is to:
  - Record user activity as a sequences of stimuli-action pairs
  - Encode these as rules
  - Incorporate these rules into competing processes as part of a BBA
- Various types (many of which use fuzzy logic):
  - **Evidential learning**
  - Case Learning
  - Instance learning
  - Medal-Wang Learning
  - ISL/AEE/AOFIS
  - etc

See: Callaghan V, Clark G, Colley M, Hagras H Chin JSY, Doctor F "*Intelligent Inhabited Environments*", BT Technology Journal , Vol.22, No.3 . Klywer Academic Publishers, Dordrecht, Netherlands, July 2004

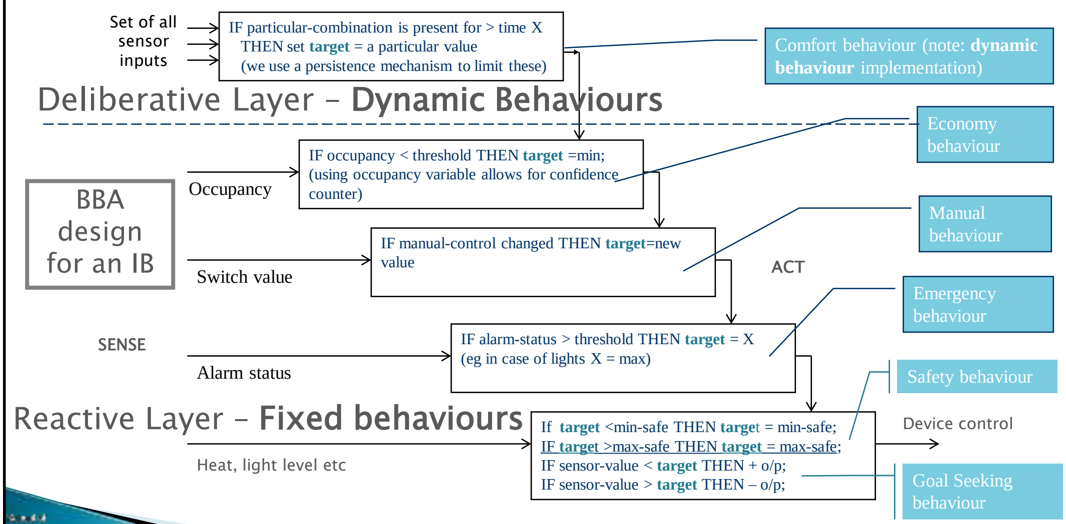
Evidential Learning

- Whenever new user event occurs, takes "snapshot" of sensors (inc data from other agents) and records in table ("*evidence-record*") after n occurrences ("*evidence inertia*")
- Continually monitors user events and "*evidence records*"
  - pre-empting proven reoccurring users events (habitual behaviour),
  - adding new events that exceed "*learning inertia*" barrier
  - merging similar "*fuzzy evidence*" records
  - removing dead or infrequently used *evidence records* based on some *evidence-significance* measurement (eg no of events, time since last used)
- Inspired by CASE learning which uses encapsulated prior experiences as basis for dealing with new situations.

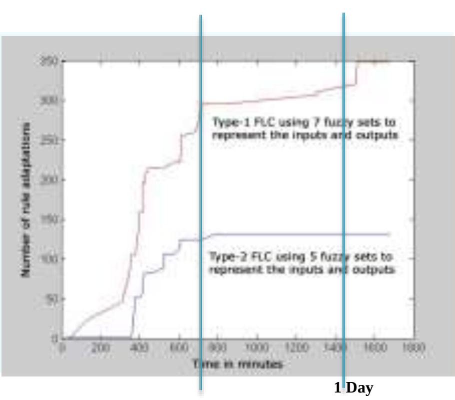
Evidential Learning

- Evidential Learning is simply another behaviour layer (but a dynamic behaviour). Thus the system now has 2 types of behaviour
  - **Fixed** – do not change or adapt over time (the standard approach in behaviour systems)
  - **Dynamic** – the agent learns by forming new instances of behaviours based on monitoring user actions (ie the deliberative component)

Example of BBA Based Evidential Learning



Example of Rules and Learning Rates



IF **InternalLightLevel** *is* **VVLOW** *AND* **ExternalLightLevel** *is* **VVLOW** *AND* **InternalTemperature** *is* **VVHIGH** *AND* **ExternalTemperature** *is* **MEDIUM** *AND* **ChairPressure** *is* **OFF** *AND* **BedPressure** *is* **ON AND Hour** *is* **Evening** *THEN* **ACTION_Light1_value** *is* **VHIGH** *AND* **ACTION_Light2_value** *is* **HIGH** *AND* **ACTION_Light3_value** *is* **LOW** *AND* **ACTION_Light4_value** *is* **VVLOW** *AND* **ACTION_Blind_state** *is* **CLOSED** *AND* **ACTION_BedLight_state** *is* **ON** *AND* **ACTION_DeskLight_state** *is* **OFF** *AND* **ACTION_Heater_state** *is* **OFF** *AND* **ACTION_MSWord_state** *is* **STOPPED** *AND* **ACTION_MSMediaPlayer_state** *is* **RUNNING**

Example of Learning Rates **Example of Generated Rules**

Evidential Learning – Some Issues

- How to differentiate between "*logically identical, but physically different*" instances (eg 21c and 21.5c)?
- How to decide what to learn (eg how to avoid erratic or one-off events that have little or no long term significance)?
- How to prevent memory being over-run (eg Life-long learning could produce a huge number of rules)

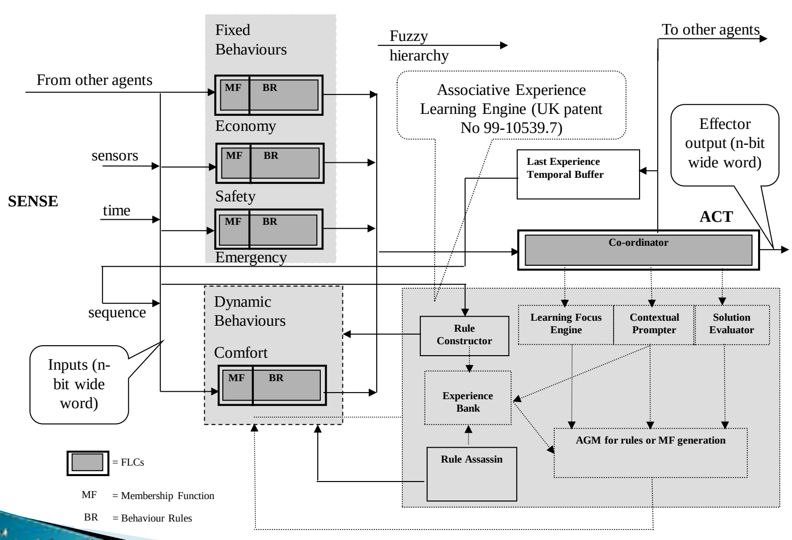
Evidential Learning – Some Features

- **Dynamic Behaviour**
  - Rule Disambiguation – new rule only created if input vector beyond some distance from nearest neighbour
  - Learning inertia – new rule only activated after persistent use (ie only if the number of examples in a period X exceeds some threshold Y – done by incrementing counter)
  - Behaviour assassination – dormant rule removal (ie remove if its use falls below threshold Y in period X – done by inc/decrementing a "*life counter*")

Incremental Synchronous Learning

- Uses **Fuzzy Logic** used to encode sensed parameters into a probabilistic membership of sets describing sensory ranges (eg temp sets might be cold, warm hot).
- **Sense-Action** rules formed based on fuzzy descriptions Fuzzy rules implemented as processes forming behaviours and arbitration control.
- **Fuzzy matching** to find appropriate, or eliminate duplicate, sense-actions.
- **Arbiter o/p** fed to other agents and back to input to broadcast compressed sensory status or give temporal action.
- Can be regarded as refinement of evidential learning

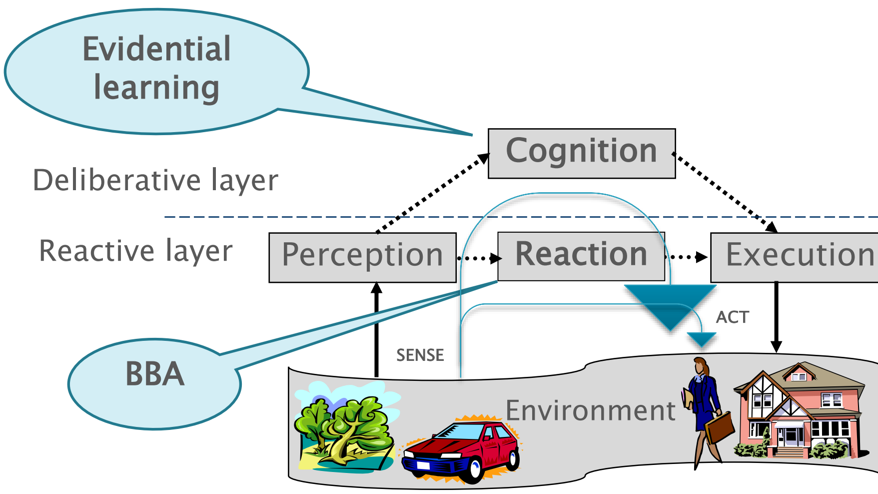
Example of BBA Based ISL Implementation



More Information:
Callaghan et-al 2000 "*A Soft-Computing DAI Architecture for Intelligent Buildings*"

You will get a more detailed explanation of Fuzzy Logic in Prof Hagras' course

Embedded-Agents – We Have a Solution



End of Part 2

That's it for Part 2 !

Part 3

- People's Views of Intelligent Environments
- A Matter of Control
- Agents Versus People (a question of balance)
- Attitudes to Autonomy
- End-User Programming
- Adjustable Autonomy
- The Singularity

School of Computer Science & Electronic Engineering
Essex University

http://victor.callaghan.info/publications/misc/2014_KAU_Training-IE_June2014.pdf

Better Intelligent Environments

- To a large degree, **people** are the "customers" of IE environments
- So, to some extent, the judgment of better Intelligent Environments is the **judgment of people**, or users.
- What are **users views**, what are the bothered about?

School of Computer Science & Electronic Engineering
Essex University

What Are People's Views?

- **Venkatesh** (2001) University of California - *attitudes to smart home technologies*
- **Chung** (2003) Samsung Corp, American Institute for Research - *smart home requirements in USA & South Korea*
- **Barkhuus and Dey** (2003) University of Copenhagen - *is context-awareness taking control away*
- **Röcker** (2004) Fraunhofer Institute, Philips Research and France Telecom - *cross cultural expectations of to smart homes in multiple European countries*
- **Mäyrä** (2006) Tampere University Hypermedia laboratory - *expectations of digital homes*
- **Montano** (2006) Goteborg University – *attitudes to smart homes*
- **Davidoff** (2006) Carnegie-Mellon University - *type of control of digital homes*
- **Rukzio** (2006) University of Munich - *interaction with technology in digital homes*
- **Chin** (2008) University of Essex - *study of user control issues in smart home*
- **Ball** (2011) University of Essex - *study on perceptions of autonomy*

- A commonality found in all these studies is that **maintaining control is a paramount concern** for potential users of IE environments.
- Additionally, issues concern adaptability, customisability and transparency of the system, as well as privacy of personal information and trust.
- The studies also found that people can balance concerns against potential benefits (eg mobile phones, energy conservation etc).

Matthew Ball, Vic Callaghan, "*Perceptions of Autonomy: A Survey of Users' Opinions Towards Autonomy in Intelligent Environments", Intelligent Environments 2011 (IE'11), Nottingham 27-29th July 2011*

Agents – A Question of Control?

"The dream of technology is the dream of control…control is an illusion; absolute control, even if it were possible, would be disaster." William Byers

- "*we have adopted an optimal control framework in which failing to satisfy each objective has an associated cost. A* ***discomfort cost*** *is incurred if inhabitant preferences are not met … An* ***energy cost*** *is incurred based on the use of electricity … discomfort is indicated by overriding the choices of <the controller> and this relative discomfort is translated to a dollar amount by means of a* ***misery-to-dollars conversion factor***" (Mozer 98)

- "*a contrasting paradigm is to see the* ***'user as king/queen'*** *and create agents that* ***'particularise'*** *(rather than generalise) to a specific user's needs, and respond immediately to whatever the end user demands (providing it does not violate any safety constraints)*" (Callaghan 04)
- "Some lay people **distrust autonomous agents** and prefer to exercise **direct control** over what is being learnt and when … or use their **creative** talents … to become designers of their own systems" (Chin 09)

Control– Who Is The User (individuals or groups)?

- In communal spaces, who chooses the shared settings

- Inspiration from companies

Groups = “collective individual”

Callaghan V, Colley M, Clarke G, Hagras H, "A Soft-Computing based Distributed Artificial Intelligence Architecture for Intelligent Buildings", In book entitled “Soft Computing agents: New Trends for Designing Autonomous Systems”, International Series "Studies in Fuzziness and Soft Computing“, (Eds: V. Loia, S.Sessa), Springer-Verlag, Volume 75, pp. 117-145, 2002

Agents versus Human Agents – A Question of Balance?

- If Technology Plays a hand in control of our environments then:
  - What is the balance between machine versus user control
  - How is that achieved?
- What do people mean by control
  - The freedom to make choices for themselves – **autonomy?**

Autonomy

no (or little) assistance from people !

A Machine View

- "*the ability of machines (or sub-entities) to usefully react to a changing and perhaps unexpected environment without continuous human control*".

"*autonomous systems are ones which govern themselves (ie develop their own rules) in contrast to automatic systems which are simply self-regulating (ie execute rules generated elsewhere)*"

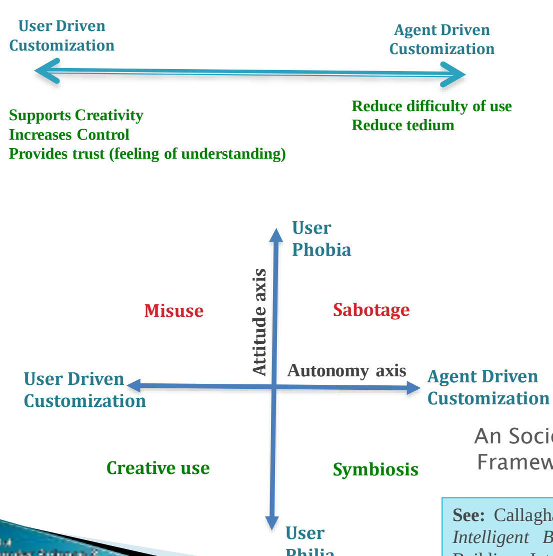
Human View

- Involves terms like "*Free Will*" – but what does that mean (and does it exist!)

Are people any more than reactive agents; "*puppets with strings being operated by the environment*" (**Sharky's puppets analogy**) – complex behaviour (intelligence?) arises from a complex world (type into Google "*the Free Will Illusion*")

- So reducing autonomy is akin to getting more assistance from people – teamwork or more commonly, Agent Teamwork

Attitudes to Autonomy



A general assumption underpinning this model is the view that the less understanding of, and control over, their technological environment that people have, the more resistant or fearful they will be of it (and vice versa).

An Socio-Agent Autonomy Framework for AmI Research

See: Callaghan V, Clarke G, Chin J "*Some Socio-Technical Aspects Of Intelligent Buildings and Pervasive Computing Research*", Intelligent Buildings International Journal, Vol 1 No 1, 2008

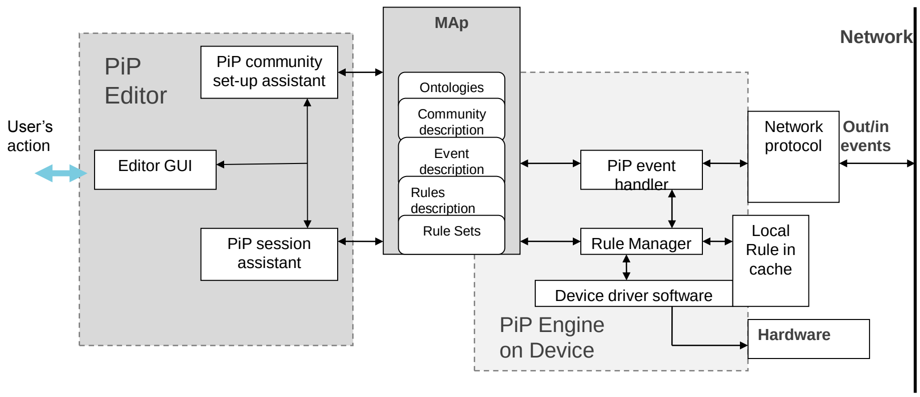
End-User Programming

- Programming-by-Demonstration (or Programming-By-Example)
  - Introduced by David Canfield Smith in the mid-seventies
  - Popularised by Henry Lieberman in nineties
  - algorithms / functionalities not described abstractly but demonstrated in concrete examples
  - Uses software agent that records interactions between user objects creating program corresponding to user's actions" [Henry Lieberman]
  - Reduces the gap between user interactions and delivered program functionality by merging the two tasks
- Main area of PBE focused on graphical user interfaces running on PCs
  - Peridot allows computer screen interfaces to be created by non-programmer users
  - "SmallStar" allows users to create programs on Xerox Star Office information system
  - Stagecast Creator a rule-based PBE model allowing children to construct a graphical grid world
  - Also has been employed in World Wide Web related technologies, computer games or text editing tools

Pervasive-interactive-Programming (PiP)

- Event-based modular architecture
- System is explicitly put into learning mode & taught (by demonstration) how to behave by the lay end-user
- Networked devices "learn" tasks by "listening" to the user interactions with networked devices
- Has the notion of communities (virtual appliances)
- Interfacing via editor (called PiPeditor) providing means to:
  - set up / amend communities (virtual appliances)
  - manage teaching session so tasks be taught (via the editor or physically operating networked devices).
- A back-end engine to generate and execute rules (PiPvm)

PiP Architecture

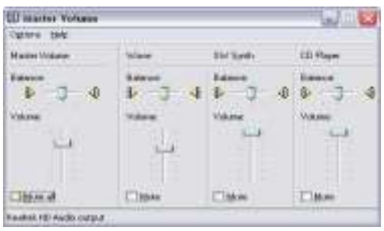


Editing Associations

- Manual Association:
  - Use PAD or smart-phone based graphical editor to manually create a list of linked devices (based on ontology or type information)
  - Physically touch (or otherwise indicate – eg switching on/off) devices to be linked (see related topic of "Tangible Computing" by, Ishii at MIT, which seeks to give data and soft components physical form)

Adjustable Autonomy

- Imagine a sliding scale switch (like a volume control) for each system in the environment.
  - So we have a theoretical mixing-desk for autonomy in the system.

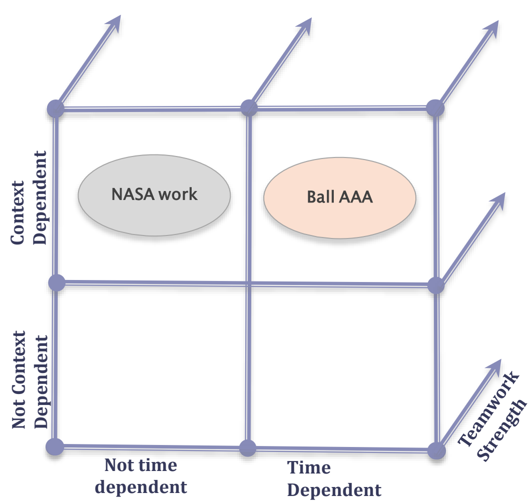


- The autonomy scale – how much control do people have?
  - 2 extremes

- End-user programming approaches
  - Empowers the user
  - Well suited to creatively minded users
  - What if user isn't able or willing to use the system?
- Autonomous-agent programming approaches
  - Reduce cognitive load placed on the user
  - Works by guessing users intentions, so prone to making wrong
  - Lack of transparency can cause distrust

Essex University

Adjusting Autonomy



Ball AAA Discrete Modes

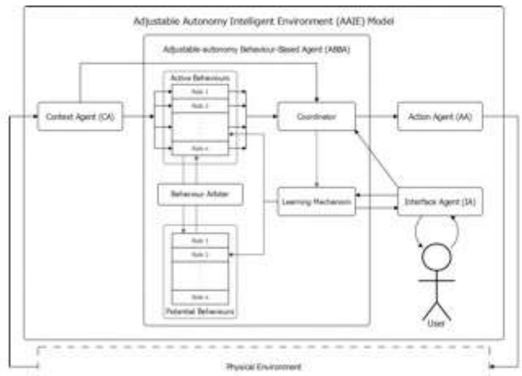
- **<u>Full autonomy</u>**: agent learns from the user's behaviour, automatically creates/maintains rules as the agent deems it necessary.
- **<u>High autonomy</u>**: agent learns rules from the user's behaviour which can only become active when confirmed by the user (agent teamwork).
- <u>Low autonomy</u>: user creates/maintains rules assisted by the agent presenting suggestions (agent teamwork)
- **<u>No autonomy</u>**: the user creates/maintains rules with no assistance from the agent.

Essex University

106

An Adjustable Autonomy Agent

- Based BBA & ISL
- 2 sets of behaviour (active & potential)
- Each rule has a 'usefulness' parameter (how frequent & accurate rule has proved)
- Variable autonomy achieved through varying usefulness threshold (a differential of standard learning inertia).



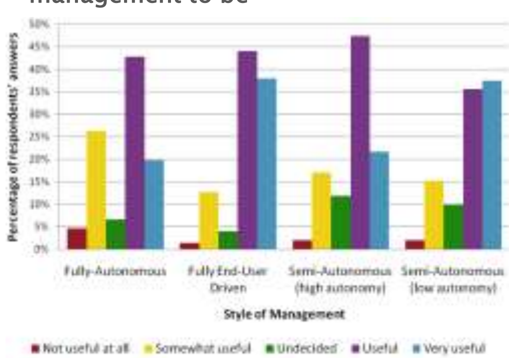
Ball et-al "*An Adjustable Autonomy Agent for AmI*", IE2010, Malaysia

"*governing a system at a sweet spot between convenience (delegating every bit of work) and comfort (delegating only what agent can be trusted to perform)*" (Bradshaw 04) ie adjustable autonomy allows agent to 'back-off' certain tasks and let user take control whenever user so wishes.

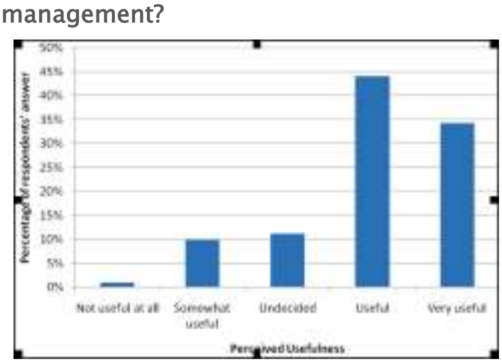
Essex University

Findings – 1

How useful the survey respondents perceived the different styles of management to be



How useful would you find the ability to change between the different styles of management?

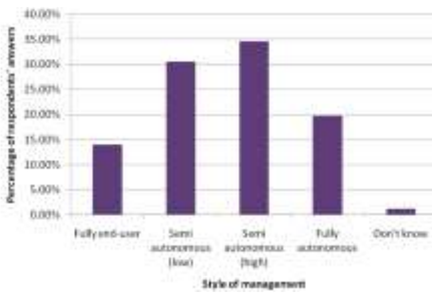


Essex University

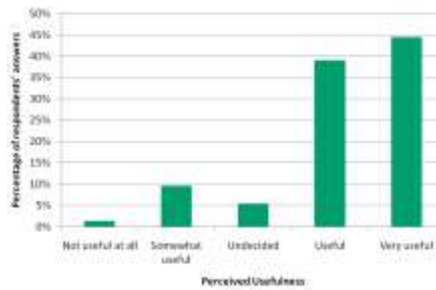
Findings – 2

Sub-system	Answer / Percentage of respondents answers				
	Fully end-user	*Semi-autonomous (low)*	*Semi-autonomous (high)*	*Fully-autonomous*	*Don't know*
Lighting	12.50%	28.95%	36.18%	20.39%	1.97%
Heating	13.16%	29.61%	39.47%	16.45%	1.32%
Entertainment	27.63%	36.84%	25.00%	10.53%	0.00%
Security	13.16%	28.95%	32.89%	23.68%	1.32%
Environmental	3.29%	28.29%	39.47%	27.63%	1.32%

How useful is it to choose the style of management for individual parts of the system



Overall popularity of the different management styles



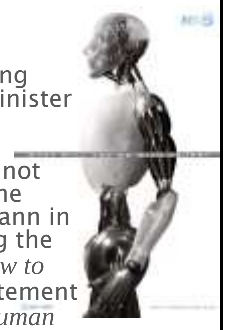
Usefulness of being able to choose the style of management for individual parts of the system

The Singularity

- The Singularity …. *the moment machine intelligence exceeds human intelligence* (around 2050 according to Kurzweil)

... Might be brought to fruition as outcome of *whole brain emulation*, *transhumanism* or an *intelligence explosion*!

Questions About The Singularity



- **Will it arise**: Do you think a singularity can arise from developing current technology; can faster machines alone usher in an age of computers that boast human-level intelligence, or are there yet to be discovered fundamental processes (eg quantum physics) involved in the brain)?
- **Will it be good**: Do you think that that machines capable of surpassing human intelligence are inherently good or will they have they more sinister consequences?
- **Further Information:** Thoughts about super-intelligent machines are not new and can be traced back to people such as Richard Thornton in the 1840s (mechanical calculators) and Alan Turing and John von Neumann in the 1950s (computing pioneers. Vernor Vinge is credited with writing the most seminal article on this "*The Coming Technological Singularity: How to Survive in the Post-Human Era*", which contained the much quoted statement "*Within thirty years, we will have the technological means to create superhuman intelligence". More recently, in the early part of this millennium, Ray Kurzweil wrote "*The Singularity is Near*" which is a comprehensive and popular account of the issues. According to Kurzweil the singularity will occur around 2045 which is broadly consistent with the views of other experts who seem to place it around 2040.

Amnon H. Eden, James H Moor, Johnny H Soraker, "*Singularity Hypotheses: A Scientific and Philosophical Assessment*" (The Frontiers Collection), Springer, April 2013, 452 pages ISBN-13: 978-3642325595

University of Essex

That's it!

vic@essex.ac.uk

More information can be found on:
http://victor.callaghan.info